LOS ANGELES ATARI® COMPUTER ENTHUSIASTS

LA ACE

NEWSLETTER

VOL. 7 NO. 10

OCTOBER 1987

October Meeting Schedule

General Meeting: Tues Oct 6th 7PM
Granada Hills High School

ST Sig Meeting: Tues Oct 6th 7PM
Mercury Savings & Loan

Future Meetings

General	ST Sig
Wed Nov 4	Tues Nov 17
Tues Dec 1	Tues Dec 15

ELECTIONS - NOV. 3rd.

Volunteer! Run for a LA-ACE Office
Call Warren Farina - (818) 893-6913
Chairman, Nominating Committee

Tell A Friend

For each new membership you bring into LA-ACE you will earn $5.00 credit towards your membership renewal plus the new member will receive 4 door prize tickets free.

Meeting Agenda

DataSoft/Intellicreations will demo
Alternate Reality - The Dungeon

Ray Maynard will demonstrate "MIDI"

Door Prizes:

DataSoft Software
"Classic" ATARI Games
Dbasic w/manual (Faster than " C "
8 bit Disks of the Month

NEWSLETTER AWARDS:
1st Prize - Tim Laren
2nd Prize - Doug Kelley

LA ACE BBS (818) 700-1652

LA ACE BBS (818) 700-1652

THE FOLLOWING JOINED OR RENEWED IN SEPT:

Jonas Robleda
Donald Foster
James White, Jr.
Gary Cuesta
Terry Andolino
Ed Davenport

Hugh Edwards
Jeanne Mannes
Edward Sheanin
Stephen Arnone
H. R. Baird
Chuck Sickler

LA-ACE Newsletter

This newsletter is written and printed by the members of the Los Angeles Atari Computer Enthusiasts, an association of individuals with a common intrest, using and programming Atari Computers. This group is not affiliated with Atari Corporation nor with any other commercially oriented organization. Any logos, tradmarks or company names are used with permission or as an informal methoud of refering to a product or company.

Opinions expressed in this publication are those of the author and do not necessarily represent the opinion of LA-ACE.

LA-ACE Membership

12 month membership fees:

Individual	$20.00
Family	$20.00
Associate	$6.00

Individual and family membership includes subscription ti this newsletter, membership access to the LA-ACE BBS, and access to the general meetings. Associate membership includes access to LA-ACE BBS only.

LA-ACE Meetings

The General Meetings are held on the first Tuesday of the month at 7:00pm unless othwise posted in this newsletter. All other Special Interest Group meetings (SIGs) times will be posted in this newsletter.

Advertising

This newsletter is supported through the members of LA-ACE and it's suporters. All advertisments must be submitted, camera read and paid in full, no later than the first Saturday following the General Meeting. There are discount for long term or concetuive advertisments.

Advertising Rates:

Full Page	$40.00
1/2 Page	$25.00
1/4 page	$15.00
Business Card	$5.00

President	Vice-President
Bill Lurie	Jon C.A. DeKeles
818-780-1723	818-363-5660
Secretary	Disk Librarian (16-bit)
Doug Kelley	Tim Laren
818-349-5006	818-366-3817
Treasurer/Membership	Disk Librarian (8-bit)
Michele Rose	Wayne Habberstad
818-700-0893	818-341-5270
Text Librarian	Equipment Librarian
Warren Farina	Ed Davenport
818-893-6913	818-884-4695
BBS SysOp	Circulation
Rich Monosson	Kam Bansal
818-700-1652 (BBS)	818-891-3171

Interim Newsletter Editors

Bill Lurie   Rich Monosson   Michele Rose

1-800-555-1212

PRINT MASTER
Unison World, Inc.
by John W. Morgan-LAACE
$39.95 List

Print Master is a graphics utility program that allows you to design and print cards, signs, banners, stationery and calendars on your 520ST or 1040ST. It also includes a graphics editor to allow you to design and save your own ideas for use in these applications. Sounds like Print Shop, doesn't it? While similar in appearance, Print Master outperforms Print Shop in many respects.

The TOS extender on the file used to run this program gives away the fact that it is an adaptation of an IBM program. The doc reinforces this view, though a slip of paper included with the manual promises a manual specific to the ST if you request it when you send in your registration card. The program is intuitive enough that this isn't necessary. Like other programs, the Print Master steps you through each section with a series of menus. The screen displays white type on a black background, wit windows opening to display graphics i black on white. The doc spends less time on instructions for the program than on hints and suggestions for designs.

The disk comes with a library of 111 drawings, 11 background patterns, 8 fonts and 11 borders. More than one graphic or typestyle can be selected for a page. Extra menu commands handle the graphic choice, while keyboard commands toggle different typestyles in addition to texture (3D, Outline or Solid), alignment and size. Both the drawings and borders show more detail and imagination than in other programs, with borders of ants or footprints used instead of just lines or tones. A disk called "Art Gallery 1" provides 140 more graphic shapes and patterns. The final menu allows you to save your design on disk to call up as a ready made item. Most of the formats for printing are just like other programs. A sign or stationery takes a single standard page, banners take six pages of continuous-feed paper and cards are printed so the page can be folded i fours to card size. No colored paper o envelopes are provided with the program, but a card suggests a supplier.

LA-ACE

LOS ANGELES ATARI COMPUTER ENTHUSIASTS
P.O. BOX 7752, VAN NUYS, CA 91409

SPECIAL EDITION

"The Best of LA-ACE"

ATARI®

COMPUTER FAIRE VERSION 2.0

FEATURING

The New ATARI Corporation

SATURDAY SEPT. 19, 1987   10-6 PM
SUNDAY SEPT. 20, 1987   10-5 PM

Los Angeles Atari Computer Enthusiasts is an association of individuals with a common interest in all ATARI computers, programming and usage. This group is not associated with ATARI Corporation, nor with any other commercial entity. Any logos, trademarks or company names are used with permission or as an informal way to refer to a product or company.

LOS ANGELES ATARI COMPUTER ENTHUSIASTS

MEMBERSHIP APPLICATION

LA ACE is an association of individuals and families with a common interest...using and programming all **ATARI** Computers. **LA ACE** membership includes a subscription to the monthly newsletter, access to the club's disk library, text library, extended bulletin board privileges and discounts at certain computer stores. We provide you with the opportunity to be with other **ATARI** owners and programmers to talk about and exchange or solve ideas and problems you may have. Our meetings are usually the first tuesday of each month except when it falls near a holiday. Then it is postponed until the following tuesday. Guests are welcome!!! Meetings are held at Granada Hills High School 10535 Zelzah Ave(between Devonshire and Chatsworth).

Dues are $20.00 per year per person, or $25.00 per year per family. Please make checks payable to LA ACE. Dues may be paid at the door, or mailed to:

LA ACE
P.O. BOX 7752
VAN NUYS, CA 91409

___ New Membership ___ Renewal
___ Individual ___ Family

Name(Print):________________________________ **Date:**__________

Additional Family Members:____________________________________

Street Address:____

City:_________________________ **State:**_____ **Zip Code:**________

Phone(Home):__________________ **(Work):**______________________

****

(For **LA ACE** use only)

Amount paid: $____________ Date received:_________________

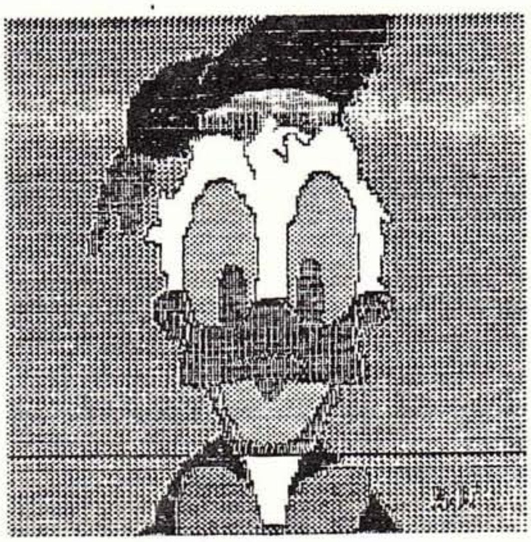
Date entered into data base:_____________ Card #:_____________

A nice addition to the standard formats is a calendar option. It allows you to print a monthly or weekly calendar in any year between 1900 and 2099. Graphics and type can be displayed at the top, while important dates can be noted within the squares or slots allotted for each date.

When you finish designing, the print option will (after a 'thinking' delay) display a print preview window which displays the finished design as it appears on the page. If things aren't quite what you wanted, you can return to the main menu and step back through the section. All of your previous choices are still there, so the part that you want to change is all that you have to reenter. Several printers are supported by drivers, including (besides Epson) C. Itoh Prowriter 8510, Star Gemini 10X & 15X and three Toshiba printers. A trouble-shooting guide lists thirty-six printers that can be used, but this was probably for the IBM version, since exotics like inkjet or laser printers are mentioned. Unison World provides both telephone and mail support which will make adapting an unusual printer easier. Their number is (415)848-6666 from 9am to 5pm PST. Your safest bet is to go by the listing on the outside of the package.

Print Master is a disk intensive program which means some delay as the files are taken from the drive for borders, fonts, graphics and printing instructions. If you have a 1MEG 520ST or 1040ST these delays can be eliminated by copying the Print Master to a RAM disk (a nice one is available from the Houston ACE, written by Michael Vederman. To get a copy, contact Tim Laren, 16 Bit Librarian) and running it from there. If you know where to go, composition times are reduced to seconds rather than minutes. My disk was not copy-protected which made things easier. Please copy this program only for use on your own system, so manufacturers are encouraged to release unprotected programs. A spool or print buffer will free up the computer for other uses quickly.

The mouse can be used on this program, but within restrictions. It's used mostly for cursor control and return functions. The restrictions are easiest to see in the Graphics Editor, where keyboard commands control most functions and are easier to



use when they can be replaced. The Graphics Editor displays x/y coordinate numbers to make predesigning an idea using graph paper easier. The editing window is 88 pixels wide by 52 tall. There are 11 commands to modify the picture with, including flipping it horizontally and vertically, inverting the image (white on black) and a gem-like window command that positions an adjustable window over a section of the design to duplicate it in another area. Also included are freehand drawing, drawing in gray (a broken line) and printing the design for a hardcopy preview. It's best to disable the keyclick with the control panel before using the Editor, since it clicks incessantly, especially when drawing freehand with the mouse.

I found Print Master to be easy to use and powerful. Very little reading was required to get the program up and running, while extra features seemed to call attention to themselves . I don't care for white type on a black background, but no provision for transforming the colors (as in ST Writer) was noticed. No additional fonts were mentioned, even in the library disk. I'd like to see more typestyles to work with. The windows that display the graphic choices or print preview are white with black letters which makes the menus seem gloomy in comparison. Otherwise, the many features and flexibility of the program made it useful immediately.

EVERYTHING YOU EVER WANTED TO KNOW ABOUT A MODEM BUT WERE AFRAID TO ASK

by Tim Laren

(This article is one of a series of "LAACE Classics", i.e., reprints of articles from earlier LAACE newsletters. This selection is from the Nov. 1982 edition, published when the club was known as the West Valley Atari Users Group. I have modified the wording slightly to modernize the terminology but the article is mostly in its original form...Ray Maynard)

What is a modem? What does it do? How does it work and what will it do for me? How do I hook it up and what software do I need to use a modem? What does a modem cost and what options do they have? Do some go faster and can I communicate with any other modem? These are just a few questions about modems and I hope to answer these and many more in this article.

First of all is "What is a modem?". The word MODEM is actually a contraction of MOdulator-DEModulator. A modem is an electronic device used to convert digital signals to an analog form to be transmitted over the telephone network. Since the telephone network was designed for analog voice transmision, it is not possible to transmit digital signals from a computer or terminal over the telephone in their present form. The telephone network can pass a signal with a frequency range of 300 to 3300Hz (cycles per second). The modem converts the digital data into an analog signal in this band. The modem can also receive this analog signal and convert it back into digital data for use by your computer or terminal.

The 103/113 type modem can transmit data in both directions at the same time. This is known as FULL DUPLEX operation. Both the transmitter and the receiver work independently of each other. In another mode, HALF DUPLEX, only the transmitter or the receiver are active at any one time. In half duplex the answering modem controls the data direction. Yet another mode, SIMPLEX, is used in some places where only one direction communication is needed. With simplex, one modem only transmits and the other only receives. Simplex is not used by the home computerist because of its lack of versatility.

Most people reading this will be interested in a 300 baud modem, i.e., a Bell 103/113 compatible modem.

I will attempt to explain its basic functions. The Bell 103/113 uses simple frequency modulation, sometimes called FSK (Frequency Shift Keyed Modulation). It transmits one frequency to indicate a digital one (1) and a different frequency to indicate a digital zero (0). This style modem transmits and receives at the same time, which means four different frequencies (tones) are needed. The modem originating the call transmits an originating tone ot 1070Hz for a "0" and a tone of 1270Hz for a "1". The modem at the answering end transmits a tone of 2025Hz for a "0" and 2225Hz for a "1".

Your computer works with eight bits at a time, so how can two tones send all eight bits over the phone line? Good question! All you do is send them one at a time! With the Atari you hook up the 850 Interface Module and it takes care of the conversion and timing. The output of the 850 is asynchronous (async) serial data. Async serial data is the same data sent out by the computer but instead of sending all 8 bits out at the same time, only one bit at a time is sent. The modems we will be discussing have a rate of 300 bits per second (300 baud).

With async transmission the start and end of each byte must be marked. When idle, the serial line is kept in the mark (1) state. To signal the start of a byte, a space (0) (1 bit) is sent, followed by the 8 bits comprising the byte. To signal the end of the byte, a stop bit (actually 1, 1.5, or 2) is transmitted. The stop bit(s) are in the mark (1) state and the serial line stays in this state until the start of the next byte. Some system use the 8th or 9th bit of the byte as a parity bit to help detect errors. The parity bit can be either set to a mark state, a space state, odd (total number of bits in the byte is an odd number), or even (total number of bits in the byte must be even). Wait a minute! What's a one (1)? Where does it go? I'm lost!

```
20 OPEN #2,4,0,"K:"
```

Open the keyboard for input only

```
30 XIO 40,#1,0,0,"R:"
```

Sets up the serial port so that we can send and receive at the same time without losing any data. (For more info read the Interface Manual).

```
40 STATUS #1,R
```

Reads the status of the serial port into the computer's memory at location 747)

```
50 IF PEEK(747) THEN 70
```

Location 747 contains the number of characters in the buffer that have been received but not processed by this program. If this equals zero, no characters are waiting so skip the receive routine contained in line 60.

```
60 GET #1,A:Print CHR$(1)
```

Get a character from the serial port's buffer and print it on the screen with no modification. A translation routine might be placed here to work better with some dial-up computers.

```
70 IF PEEK(764)=255 THEN 50
```

Check if there is any keyboard input. If not, go back to check for more serial port input.

```
80 GET #2,A:POKE 764,255:PUT #1,A
```

Get a character from the keyboard, clear the keyboard buffer and put the character out over the serial port.

```
90 GOTO 50
```

Go check the serial port again.

That's all you need to get started!

(I have deleted a section discussing accoustic modems here. Accoustic modems are almost unheard of these days and they never were very reliable anyway. Most modems sold today are of the direct-connect variety discussed below. You can get some 300 baud models for well under $100 and 1200-baud modems can be had for as low as $250. 2400-baud modems are already being sold but are still a bit expensive for most home computer owners...Ray Maynard)

Direct connect modems have electronics that allow them to connect directly to the phone line. Some direct connect modems also have features like auto-answer and auto-dial-out.

When you start looking at direct connect modems you will find two major types, dumb and smart. The "dumb" modem may allow you to dial out or answer under the control of your computer or terminal, but with the "smart" modem you can do this and more under control of the modem itself (that's why it's "smart").

In the dumb direct connect modem you should find status lights and dial-out/auto-answer modes (dial-out lets your computer send pulses to the phone line to dial; auto-answer connects the modem after a ring has been detected on an incoming call). Some direct-connect modems require no extra power. These devices get power from the phone lines and so you don't need an extra plug on your wall. This is a handy feature. For most direct connect modems a modular phone plug is required (all new "state of the art" modems have them so as to get FCC approval) so make sure you have one near your computer. Another handy item is a monitor speaker that allows you to hear the modem working (dialing and data transfering). It is also nice to be able to turn off this monitor if it gets to be a "pain" after several hours.

Last, but not least, is the smart direct connect modem. There are many features like five-number memory, auto redial on busy, programmable rings before answer, and dial another number if the first one is busy or no answer. Most of these features can be duplicated very easily with a little software, so you must decide if you want to get software or the smart modem. If you are using the modem on a computer terminal all the smart features are very important because the terminal can't do them. But your computer can! So think before you buy! The "smart" part of the modem can cost anywhere from $100 to $300 additional.

Well, in data communications there are standards, like RS232. Ever hear of it? The EIA (Electronic Industries Association) standard RS-232 defines the interface between things like modems and computers. The EIA standard does not define what data is transferred, just the electrical and physical specifications. Some of the physical specifications concern things like the connector, a DB-25. The male connector is used on the data communication equipment. The logical voltage levels, +5 to +15 for a space (0) and -5 to -15 for a mark (1) are also specified. Here is a table of the basic signals used in connecting a modem to a computer as defined by the EIA RS-232 specification:

EIA RS-232 PINOUT

PIN	EIA	DESCRIPTION
1	AA	Protective ground
2	BA	Transmitted Data
3	BB	Received Data
5	CB	Clear to Send
6	CC	Data Set Ready
7	AB	Signal Ground
8	CF	Carrier Detect
20	CD	Data Terminal Ready
22	CE	Ring Indicate

Only the necessary signals were made available on the Atari Interface Module. These signals are available on the 9-pin "D" connector as described below:

ATARI 810 INTERFACE

PIN	EIA	Description
3	BA	Transmitted Data
4	BB	Received Data
7	CA	Request to Send
8	CB	Clear to Send
6	CC	Data Set Ready
5	AB	Signal Ground
2	CF	Carrier Detect
1	CD	Data Terminal Ready

Let's connect a modem to an Atari. What do I do? Do I connect Transmitted Data of the computer to Transmitted Data of the modem. or what?

Well, to start out, Transmitted Data is an output on all devices, so, if we connect them together they wouldn't "hear" each other. We have to connect inputs from one to the outputs of the other and vice versa. Now they all can send and receive. Below is a cable chart showing how to connect the pins between an 850 Interface and a modem.

ATARI-TO-MODEM

DB-9 pin	Signal	DB-25 pin	Description
1	CD	20	Data Terminal Ready
2	CF	8	Carrier Detect
3	BA	2	Transmitted Data
4	BB	3	Received Data
5	AB	7	Signal Ground
6	CC	6	Data Set Ready
7	CA	4	Request to Send
8	CB	5	Clear to Send
9	n/c	-	(no connection)

OK, so I burned mv fingers a little but I got my cable built. Now what do I do? I think I need to buy HomePak or some $100 program just to use my modem. Maybe Santa will bring me one.

Yeah, keep it on your list, but for now let's write a very simple program to test out the cable and we can even use it on some BBSs in the S.F Valley. The program will not support anything except sending whatever you type on the keyboard out to the modem and displaying what is received from the modem on the screen.

The first thing we need is an AUTORUN.SYS program that loads the 850 driver into memory on bootup. There is a copy in the disk library. If you don't have a disk drive you won't need this because the Operating system does this for you. (Note: the AUTORUN.SYS file on ATARI DOS 2.0S master disks will do this as will the files RS232.COM on DOS-XL disks...Ray).

When you turn on your computer you should hear a beep just before the READY appears. This is the 350 loading. If you have a disk this happens just after DOS is loaded (quick beeps) and just before the READY is shown. To test if everything is working just type

```
  OPEN #1,8,0,"R:"
```

If you get an error message check to see if the 850 is on and connected and that AUTORUN.SYS for the 850 is on the disk.

Now let's start the BASIC part:

```
10 OPEN #1,13,0,"R:"
```

This opens the first serial port with its default parameters (full duplex, 300 baud, 8 data bits, 2 stop bits).

COLDSTARTING CAN HELP YOUR COMPUTER LIVE A LONGER LIFE

Wayne Habberstad - LA-ACE

Are you zapping your computer to death? Whenever you turn on your computer, a surge of power shocks its internal components, especially its power supply. While the computer can stand repeated "zaps" over a few years, they do eventually do the poor computer in.

Turning on and off a computer is inevitable; but you can minimize it by fooling the computer into thinking it has just been turned on. When an Atari 8 bit computer is turned on it clears all of its RAM memory (except the extra RAM that lies under the ROM in XL/XE computers and the extra RAM in the 130XE) and sets itself up ready to accept programs.

The routine in the Atari's operating system that does this chore, called coldstarting the computer (as opposed to warmstarting which is simply the resetting of the computer, usually by the RESET key) can be called by you from BASIC or DOS when warmstarting the computer is not enough to clear an unwanted program from memory, or the garbage created by it. To call the routine from BASIC, use this statement: A=USR(58487). When you press the return key the computer will blank out the display and act as though it had just been turned on.

With an XL/XE computer you can coldstart it by first using the statement BYE (or the abbreviation B.), then pressing the RESET key when the self test menu appears. The RESET key coldstarts the computer in this instance because the self test routine had inserted a one into location 580, which is checked when the RESET key is pressed; any other number tells it to coldstart. You can use the POKE statement in BASIC to set location 580: POKE 580,1. Then the RESET key will coldstart the computer when pressed.

If you are not in BASIC, this is how to coldstart the computer in ATARI DOS: use menu option M (run address), then use $E477 (the hexadecimal equivalent of 58487) as the address.

Having to clear our old programs, and their results, is not the only thing to tempt you into turning off, and on, your computer. Getting boot errors on your screen after you put the wrong disk in the drive may lead you to think that turning off the computer (or the disk drive) is the only way to stop the drive from spinning the disk inside. And pressing the RESET key is not enough as the computer will again uselessly try to boot up with the wring disk. Thi si the way around it: hold the START key down and press the RESET key. The computer will think that you want it to boot up from a cassette instead of a disk (the beep that you will hear is its signal to you to ready a cassette [;auer). The disk drive will stop spinning the disk. After you replace the disk in the drive press the RESET key. The computer will forget about trying to boot from a cassette and will boot from the disk.

Your Atari is the best 8 bit computer ever made. It deserves to be well taken care of.

FOR 130XE OWNERS ONLY

For 130 XE owners there is a bonus if they coldstart their computers: programs and data on the ramdisk are safe from the coldstart process as long as there is no RAMDISK.COM on the boot disk. As mentioned earlier the coldstart routine in the operating system does not touch the extra 64K in the 130XE. But the RAMDISK.COM will reformat the ramdisk and wipe out everything on it. As the RAMDISK.COM is needed to set up the ramdisk when the computer is turned on, 130 XE owners should keep two copies of their boot disks: one with the RAMDISK.COM to use when turning on their computers, and the other without it to use when merely coldstarting them.

After a 130XE is coldstarted without a RAMDISK.COM on the boot disk, the ramdisk can be read from and written to normally except for one thing: if you try to go to DOS from BASIC then the computer will attempt to load the DUP.SYS file from Drive 1 instead of from the ramdisk. The way to remedy this is to POKE 5439,56. This will tell the computer to lead DUP.SYS from the ramdisk.

COMMONLY NEEDED ST BASIC SUBROUTINES

by Ray Maynard - LA-ACE

Some tasks seem to come up over and over again when writing programs. Some examples are clearing the screen, waiting for the RETURN key to be pressed, etc. In the hope that my solutions to some of the more common problems may be useful, I will present some code fragments that I use.

Before starting, let me say that I do not claim to be an "expert" programmer; I'm sure that there are better ways to do some of the things I present here...if so, I encourage those with better techniques to publicize them (we could all use the help!).

To clear the OUTPUT window:

```
ClearScreen: ' Clears entire OUTPUT window...
clearw 2
return
```

Note that I have omitted the line numbers in the interest of conserving space. You must, however, include line numbers in your programs.

What if you only want to clear a few lines in the OUTPUT window, not the entire window? I use this:

```
ClearLines: ' Clear lines on the OUTPUT window.
' Clears lines fln% through lln%.
' "fln%" and "lln%" stands for
' "first line number" and "last line number"
' respectively.  Line numbers range from 0
' (the top line of the window) to 19 (the last
' line of the window).
if fln%  lln% then goto CLEARX
if fln% = lln% then gotoxy 0,fln%: ? spc(72): goto CLEARX
if fln%=0 and lln%=19 then gosub ClearScreen: goto CLEARX
if fln%  0 then fln%=0
if lln%  19 then lln%=19
for cl1%=fln% to lln%
gotoxy 0,cl1%: ? spc(72)
next cl1%
CLEARX: return
```

And if you just want to clear one line:

```
ClearLine: ' Clear line "fln%" on the OUTPUT window.
lln%=fln%: gosub ClearLines
return
```

I often want to display data on the screen and then wait until the user presses a key before continuing. Although the following routine displays "Press RETURN to continue.", it accepts ANY keypress. Two keys are treated specially: If either the Esc key or control-G is pressed, a STOP statement is executed.

```
PAUSE: ' Pause until any key is pressed...
'      STOP if ESC or control-G.
?: ? "Press RETURN to continue."
pause% = inp(2)
if pause%=27 or pause%=7 then stop
return
```

How about when you display a question that requires either a Y (for yes) or N (for no) answer? Here's an example of what I use:

```
? "Do you really want to quit (Y or N): ";: gosub getYesNo
getYesNo: ' Get either a "Y" or an "N" from the keyboard
'          If neither "Y" nor "N" is entered (either
'          upper or lower-case is accepted), ring the
'          bell wait for another keystroke.
'          NOTE: calls subroutines "BELL" and "toUpper".
gYN1: x$=input$(1): gosub toUpper
if x$"Y" and x$"N" then gosub BELL: goto gYN1
? x;
return
'

BELL: ' Ring the bell.
? chr$(7);
return
'

toUpper: ' Converts all lower-case characters in x$
'         to upper case.
toUpper1% = asc("a") - asc("A")
for toUpper2%=1 to len(x$)
toUpper3% = asc(mid$(x$,toUpper2%,toUpper2%))
if toUpper3%asc("a") then goto toUpp2
if toUpper3%asc("z") then goto toUpp2
mid$(x$,toUper2%,toUpper2%) = chr$(toUpper3%-toUpper1%)
toUpp2: next toUpper2%
return
```

Notice that subroutine getYesNo calls two other subroutines, "BELL" and "toUpper". BELL does the obvious...it rings the bell (of course you have to have the speaker volume turned up a little to hear it). toUpper converts all lower-case letters in string x$ to upper case. It's used here in case the user types a lower case "y" or "n".

I use subroutine getDate to input a date from the keyboard. This subroutine verifies that the date is correctly formatted and has a "reasonable" value.

```
getDate: ' Returns date entered from keyboard in string
"date$".
goto getD1
badDate: gosub BELL
```

```
.D1: input "",date$
en(date$)8 then goto badDate
ıı mid$(date$,3,1)"/" then goto badDate
if mid$(date$,6,1)"/" then goto badDate
for date1% = 1 to 8
if date1% = 3 or date1% = 6 then goto getD2
if mid$(date$,date1%,1)"0" then goto badDate
if mid$(date$,date1%,1)"9" then goto badDate
getD2: next date1%
getDate1% = val(mid$(date$,1,2))
if getDate1% or getDate1%12 then goto badDate
getDate2% = val(mid$(date$,4,2))
if getDate2% or getDate2%31 then goto badDate
if getDate1% = 2 and getDate2%29 then goto badDate
getdate1% = val(mid$(date$,7,2))
if getDate1%F or getDate1%99 then goto badDate
return
```

The above code could be improved by checking for exceeding the maximum number of days in ch specific month but I chose not to to keep it nple. If it's important to you, it should be easy to add more checking.

And finally we have a subroutine to determine the current screen resolution. I didn't write this one...it's from a magazine somewhere but I'll never remember which one. For text displays, knowing the screen resolution is important because the different resolutions have different maximum numbers of characters per line (among other differences). It's pretty hard to do something like centering a line on the screen when you don't know how many characters wide a screen is!

```
GetRez: ' Returns an indicator of the current screen
'       resolution.
' On exit, REZ% = 1 if high resolution.
'           = 2 if medium resolution.
'           = 4 if low resolution.
REZ% = peek(systab + 0)
urn
```

This concludes my discussion of some commonly needed subroutines. I hope that some of them will be useful to you.

PROTECT YOUR COMPUTER
by John Carpenter

Editor's note: This is the first in a series of "LAACE Classics", i.e., articles that originally appeared in this newsletter in the ancient past. The following article was printed in the March 1982 issue of the West Valley Atari Users Group Newsletter (the WVAUG later grew into LAACE). The author, John Carpenter, is a former editor of this newsletter.

Don't look now, but those friendly power lines have voltage transients that can bite your computer. With commercial labor repair costs at $40.00 an hour, you need protection and protection can be bought at a price.

The most obvious source of power line transients is lightning, and while a direct strike will likely override any protective device, these strikes induce power line transients that can travel for miles. Power line transients can also be generated whenever a heavy load or short circuit occurs in the nearby power circuit. Some examples include windstorms, earthquakes, a car accident involving a nearby power pole, and any inductive load that switches on and off. Your own use of drill motors or sabre saws are examples of switched inductive loads. Then there is the unthinkable, an accidental nuclear explosion in outer space. Yes, you need protection.

The question is, at what price do you get protection and how effective is the protection that you have purchased? If you are hardware oriented and like to do it yourself, you can be more confident that you are getting the most for you money by buying the necessary parts and assembling a protection device yourself.

In my work in aerospace projects, I have identified three manufacturers who make surge protectors for electronic equipment which are applicable to 60 hz power lines at 120 VAC. As is often the case, the simplest and least expensive component may very well prove to be the most effective in limiting the amplitude of the power line voltage.

Part number C62-350L made by C.P. Clare of Chicago is a small gas discharge Surge Arrester with a 300 ampere breakdown rating. It can be installed behind your convenience outlet across the power lines, or it is small enough to be installed in a male plug. If you choose the later, a one-half inch cable access opening must be used to allow easy installation. After installation, use RTV or other electrical sealant on the access hold to prevent small fingers from touching the A.C. voltage. Parts cost on this protector is under $5.00. The manufacturer certifies that it is REA approved for standard duty arrester.

A second alternative is part number 60KS200C made by General Semiconductor Industries of Tempe, AZ. This bidirectional silicon transient suppressorsor had a trade mark name of Transzorb and is certified to exceed the requirements of MIL-STD-1399 which is Navy shipboard requirement. This part can be installed in a standard electrical utility box with a cover plate. If installation in the wall is not your bag, a line cord connection to plug in to your convenience outlet will suffice. Parts cost is near $100.00. In fact, my first choice is a combination of the C62-350AL and the 60KS200C which has the higher power dissipation at a lower breakdown voltage to provide a two step transient protection!

A third alternative is a Metal Oxide Varistor made by General Electric. I am not current on part numbers and prices.

Where can you buy parts? Try to convince your favorite computer store to stock them or try calling elctronic parts houses.

Here is some test data on the Radio Shack Spike Suppressor at $9.95 as well as the Transzorb and the gas filled Surge Arrester. The Radio Shack unit starts to limit the peak A.C. voltage at 187 VAC RMS. The rating of 500 volts at 15 ampere surge does indicate that further voltage increase can be experienced in the circuit in a manner similar to the Transzorb, which has a much higher power rating.

The Transzorb, which has a 335 volt clamp (limit) at 180 ampere surge, starts to limit at a lower terminal voltage of 145 VAC RMS. The Transzorb also has a 10 watt steady state power dissipation rating. The Clare Surge Arrester activates at 202 VAC RMS and drops to a lower terminal voltage until the first AC zero crossing occurs. This unit activates on positive or negative peaks, as do all units tested.

ANIMATION WITH THE JOYSTICK

by Ed Gombert

One of the first things I discovered while playing around with my computer was how simple it was to use joysticks in a program. They are really nothing more than IF-THEN statements.In other words, "IF" the joystick is in this position "THEN" do this. To show how this works I have written a program that uses the joystick to animate a three dot "caterpiller" similar to the centipede game. As simple as the program is I will describe it briefly.

Line 10 starts out by clearing the screen.POKE 752,1 gets rid of the cursor.POKE 82,0 sets the left hand margin at 0 instead of 2 where it usually is. the three remaining POKEs set the color for the background, border and characters.

Line 20 sets a starting value for each section of the caterpiller. lines 30 thru 70 position and print each section. Lines 60 and 70 are blanks used to "erase" the end of the caterpiller. I found that if I moved the joystick around too fast one blank didn't do the job.

Line 75 gives each section of the caterpiller the value of the section just in front of it. In lines 90 thru 160 the first dot is the last section to be given a new value depending on which direction the joystick is pointing. STICK(0) is the first joystick port and the following number indicates the stick position as shown in the diagram below.

Lines 170 thru 200 will send each section of the caterpiller to the opposite edge of the screen as they go off the sides, top or bottom. And of course line 210 sends the program back to line 30 to reposition the caterpiller based on the new information from the joystick.

CATERPILLER PROGRAM

```
10 ? "↰":POKE 752,1:POKE 82,0:POKE  709,12:POKE  710,20:POKE
712,18
20 A=3:B=0:C=2:D=0:E=1:F=0:G=0:H=0:I=0:J=0
30 POSITION A,B:? "●"
40 POSITION C,D:? "●"
50 POSITION E,F:? "●"
60 POSITION G,H:? " "
70 POSITION I,J:? " "
75 I=G:J=H:G=E:H=F:E=C:F=D:C=A:D=B
80 IF STICK(0)=15 THEN 80
90 IF STICK(0)=7 THEN A=A+1
100 IF STICK(0)=13 THEN B=B+1
110 IF STICK(0)=5 THEN B=B+1:A=A+1
120 IF STICK(0)=9 THEN B=B+1:A=A-1
130 IF STICK(0)=11 THEN A=A-1
140 IF STICK(0)=14 THEN B=B-1
150 IF STICK(0)=6  THEN B=B-1:A=A+1
160 IF STICK(0)=10 THEN B=B-1:A=A-1
170 IF A>38 THEN A=1
180 IF A<1 THEN A=38
190 IF B>22 THEN B=0
200 IF B<0 THEN B=22
210 GOTO 30
```

```
\       14
  10    |    6
     \  |  /
      \ | /
 11---15----7
      / | \
     /  |  \
   9    |    5
       13
```

STICK function values

A COMPUTER BENCHMARK

by Ray Maynard

In a rare burst of enthusiasm, I recently took it upon myself to compare the execution speeds of the several programming languages I have available to me on my Atari 800. The results may be enlightening to anyone trying to decide on a programming language. I used the "Prime Number Sieve" benchmark printed in the January 1983 issue of BYTE magazine. The program finds the first 1899 prime numbers, i.e., numbers that have no factors other then itself and 1. This program has been run on almost every computer made and seems to have become a "classic" benchmark program.

The program as published in BYTE uses an array of 8191 numbers. Since each element of a numeric array in Atari BASIC and all compatible BASICS (except Microsoft) take 6 bytes per element, the array alone requires over 49K bytes! And that's without the program! Since the only values stored into the array are 1 and 0, I decided I could do the same thing using a character string DIMensioned 8191 and storing the characters "1" or "0" into it. Since each character in a string takes only takes one byte, that saved almost 41K bytes, allowing the program to fit into my Atari. I used two methods of storing and retrieving values to/from the string: substring substitution and PEEKing/POKEing directly into the string. An example of substring substitution is:

```
   100 DIM ARRAY$(8191)
   110 ARRAY$(100,100)="1"
   120 ARRAY$(101,101)="0"
```

while an example of the PEEK/POKE technique is:

```
   100 DIM ARRAY$(8191)
   110 LOCARRAY=ADDR(ARRAY$)
   120 POKE LOCARRAY+99,ASC("1")
   130 POKE LOCARRAY+109,ASC("0")
```

I have included the timing results from a few other computers to indicate how the Atari matches up. I ran the timings of ACTION!, C65, Atari BASIC cartridge, and BASIC XL: all the other times shown are from magazine articles. The execution times shown are in seconds for 10 iterations of the main loop of the program. You will notice that executing with the screen display disabled increases the speed. This is because the CPU has to spend significant time keeping the screen updated, even when the display isn't being changed.

Computer and Language	Execution Time
Cray-1	0.110
Control Data CYBER 760, FORTRAN	0.723
VAX-11/780, UC Berkeley C compiler	1.42
Z-80 C compiler	14.0
Atari, ACTION! compiler, screen disabled	15.1
Atari, ACTION! compiler, screen enabled	21.9
IBM PC, C compiler	22
IBM PC, FORTH	70
Atari, BASIC XL, "fast" mode, PEEK/POKE method, screen disabled	101.0
Atari, BASIC XL, "normal" mode, PEEK/POKE method, screen disabled	135.7
Atari, BASIC XL, "fast" mode, PEEK/POKE method, screen enabled	148.6
Apple II, Apple Pascal	160
Apple II, FORTH	190
Atari, BASIC XL, "normal" mode, PEEK/POKE method, screen enabled	196.5
TRS-80 Mod II, UCSD Pascal	274
Atari, OSS C65 C compiler	285.7
Atari, BASIC XL "fast" mode, substring method, screen disabled	1261
TRS-80 Mod II, BASIC	1430
Atari, BASIC XL "normal" mode, substring method, screen disabled	1583
Atari, BASIC XL "fast" mode, substring method, screen enabled	1847
Apple II, Integer BASIC	1850
IBM PC, BASIC (integer)	1950
Atari, BASIC XL "normal" mode, substring method, screen enabled	2299
Atari, Atari 8K BASIC cartridge, POKE/PEEK method, screen disabled	2377
IBM PC, BASICA (floating point)	2400
Atari, Atari 8K BASIC cartridge, substring method, screen disabled	2416
Apple II, Applesoft Basic	2806
TRS-80 Mod III, BASIC	2880
Atari, Atari 8K BASIC cartridge, POKE/PEEK method, screen enabled	3492
Atari, Atari 8K BASIC cartridge, substring method, screen enabled	3543
TI 99/4A, TI-BASIC	3960

If the newsletter editor has room to print them, I have submitted listings of some of the programs that I ran so that you can compare the coding techniques used in the various languages. In the BASIC programs, a loop is used to set all elements of the array to "true" or 1. It would have been much faster to use the code sequence

```
FLAGS$="1":FLAGS$(8191)="1":FLAGS$(2)=FLAGS$
```

but I didn't in order stay as close as possible to the original pragram as presented in BYTE.

I had lots of fun (and frustration) running these benchmarks. I hope the results are of interest to you.

THE PRINT SHOP
by Broderbund
a Review
by Geoff Campbell

The Print Shop is one of the best pieces of software I have seen for the Atari computers. It is incredibly easy to use. One need never look at the documentation, due to the excellent menus and icons. The same keys are used for the same functions throughout the program. In five minutes one is completely comfortable with the program.

The main menu comes up with the following:

Greeting Card
Sign
Letter Head
Banner
Screen Magic
Graphic Editor
Setup

You may do any of the above. If you select Letter Head, sub-menus appear. First you pick the font you would like to use. Not only is the title of the font displayed, but a graphic representation of the font is drawn(very quickly) on the screen. After selecting a font, the graphic symbol to be printed is picked. Menus with at least 50 different graphic icons are presented. The icon wanted is selected and you decide where on the page you would like to place it and how many. Next comes the print menu. Number of copies may be selected, as well as a test for paper positioning.

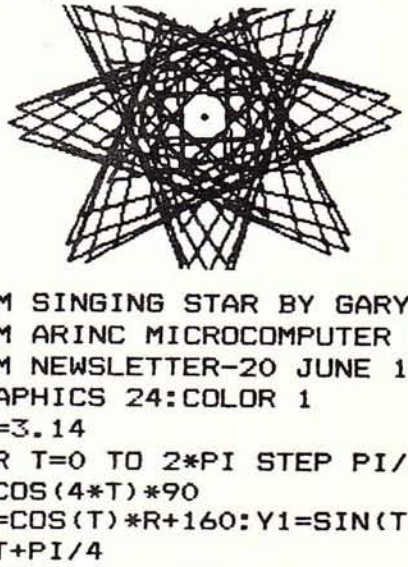
Greeting cards, signs, and banners all have similar, very friendly menus. There is always a chance to go back to the previous screen simply by pressing "ESC" or forward by pressing "Return".

The program also allows the inclusion of your own art work, which can be designed with the Graphic Editor and then saved to disk or printed. Like the rest of the program, the graphic editor is very easy to use and no documentation need be consulted.

Overall, The Print Shop is an excellent program. There are a few minor flaws with it, as there are with any program. It would have been nice if Broderbund had allowed the option for using two drives rather than having to flip the disk over from time to time. Also, the program checks for the presence of a "Happy drive" so Happy owners must run "Slow Down" to use the program. This slows down the disk access which occurs fairly often in the program. One other minor complaint might be that the inclusion of pictures from other graphic programs seems to have been ruled out. For example, neither Koala Pad nor Micro-painter pictures may be loaded into the program. These complaints are very minor, however, when the overall program is taken into consideration. The rest of the program is so outstanding that it more than makes up for minor oversights.

After seeing this program, it becomes clear to me that someone spent a great deal of time in the writing of this program. This kind of craftsmanship is really appreciated when so much of the other software on the market is unoriginal and substandard. This is the kind of software needed for the Atari to rid the "it's just a game machine image".

SINGING STAR



```
100 REM SINGING STAR BY GARY PRUITT
110 REM ARINC MICROCOMPUTER CLUB
120 REM NEWSLETTER-20 JUNE 1984
150 GRAPHICS 24:COLOR 1
160 PI=3.14
170 FOR T=0 TO 2*PI STEP PI/40
180 R=COS(4*T)*90
190 X1=COS(T)*R+160:Y1=SIN(T)*R+95
200 A=T+PI/4
210 R2=COS(4*A)*90
220 X2=COS(A)*R2+160
221 Y2=SIN(A)*R2+95
230 PLOT X1,Y1:DRAWTO X2,Y2
240 SOUND 0,ABS(X1-X2),10,8
250 NEXT T
260 POKE 77,255
270 SOUND 0,0,0,0
280 FOR I=1 TO 5000:NEXT I
290 POKE 77,0
```

****
ATARI ARTICLE INDEX
by Dave Elwood
****

Several years ago I began a list of interesting tutorial type articles written about our favorite computer. Originally I published this list in the WLAAUG Newsletter. Ive been encouraged to update it, and since Im now a member of LAACE here it is.

The list is certainly not complete for any one subject but represents a smattering of many of the more interesting capabilitys of the ATARI. The listing you see here is exactly the way I entered them complete with my own abbriviations.

Ive also included some of the best authors names in parenthesis. A few games are listed for their tutorial value.

MAG	DATE	PAGE	SUBJECT
			******* 1981 *******
COMPUTE	Sep81	67	PM GR Memory
"	"	82	Dynamic Animation
COMPUTE	Oct81	98	Extending PM GR (Animation)
COMPUTE	Nov81	96	M/L CIO in 4 parts thru Feb82 (Wilkenson)
COMPUTE	Nov81	20	Filesearch & db system
"	"	108	Numbers that make CENTS
"	"	114	Textplot
COMPUTE	Dec81	54	Maze Generator (Short & clever)
"	"	108	Blinking Characters
"	"	117	PM GR Vertical Motion
CR.COMP	Dec81	300	DLI Colors and Antic
BYTE	Jan81	18	Intro to Atari GR (Crawford)
BYTE	Sep81	284	The Atari Tut (pt1 of 8 thru Apr82)
INT	Mar81	56	Sound in Basic (Pt 1 of 6 thru Aug81)
MICRO	Dec81	45	DLI and VBLANK(Pt 1)
			******* 1982 *******
COMPUTE	Jan82	110	Superfont char Editor
COMPUTE	Feb82	19	Rounding
"	"	80	Tokens,About
"	"	90	PM GR
"	"	95	Clear Memory
"	"	77	M/L GR Routines (CIO)
"	"	100	More Display List
"	"	112	Flashing Word
COMPUTE	Mar82	30	$9.2 to $9.20
"	"	130	M/L VBI
"	"	156	Disk Tutorial
"	"	201	Screen Save
COMPUTE	May82	101	PM GR Vertical motion
"	"	112	Modens
COMPUTE	Apr82	26	Moving averages
"	"	130	Modems
"	"	136	Note and Point
"	"	146	Textplot for Animation
"	"	153	System Timers
COMPUTE	May82	72	Geomertry

"	"	101	PM GR Vertical motion
"	"	112	Modems
"	"	[illegible]	Disk speed Prog [illegible]
"	"	143	GOSUB For speed
COMPUTE	Jun82	115	Text Window in GR 0
"	"	120	Self Modify PM GR Utility
COMPUTE	Jun82	64	GR to Printer
COMPUTE	Jul82	166	M/l Fill
"	"	106	Microdos (explained)
COMPUTE	Sep82	142	GTIA (and P151)
COMPUTE	Nov82	100	USR,Getting most of
"	"	132	Arrays
"	"	164	Screen Save & Hat program
"	"	209	Adding Basic comm (Wedge)
"	"	221	XIO (Fill)
ANALOG	#6	24	DLI to twinkle starfield
"	"	33	Autorun.sys (M/L)
"	"	46	Variable lister
ANALOG	#7	13	Buncrush Basic var Utility
"	"	32	Fast M/L Char set move
"	"	34	Sound Utility
"	"	41	Multiprocessing (VBI)
ANALOG	#8	19	Disk structure tutorial
"	"	36	Mix CTIA/GTIA GR
"	"	57	Graphic Violence (Hudson)
ANALOG	#9	55	FORTH DOS in Basic
"	"	33	Disk Tool and tut (in M/L)
ANTIC	Jun82	15	Move it-M/L to move a byte
"	"	6	Modems
ANTIC	O/N82	10	Sound affects
"	"	41	Speed demon (good doc)
"	"	56	FORTH Turtle GR
CR.COMP	Apr82	115	Using Disk PT 1
CR.COMP	May82	180	Using Disk PT 2
CR.COMP	May82	30	Ramdisk review
CR.COMP	May82	184	Error 164 recovery
MICRO	Jan82	55	DLI and VBLANK (Pt 2)
MICRO	Nov82	96	Extra Colors (by Mixing)
BYTE	Nov82	194	Animation with Color Registers

******* 1983 *******

COMPUTE	Feb83	80	Fine Scrolling
"	"	98	Page Flipping
"	"	154	Superfont Plus
"	"	216	M/L utility-stringing code
COMPUTE	Mar83	150	Disk Access-Save screen
"	"	187	Mass Line Delete (8 line prog)
"	"	198	M/L-Numeric input
"	"	214	Sound tut-AUDC1,AUDF1
"	"	231	Manage memory (Brannon)
COMPUTE	Apr83	94	Visicalc cashflow mgr.
"	"	136	M/L tut Multiplying (pt2)
"	"	170	Vidio 80 (Brannon)
"	"	202	DLI Rainbow Colors
COMPUTE	May83	156	M/L Numeric output (pt 1)
"	"	166	Basic sorting
"	"	192	Wedge-Screen print (M/L)
COMPUTE	Jun83	175	PM GR Simplified

"	"	188	M/L Fragment (Crawford)
"	"	206	Deferred EDIT (Leeman)
"	"	219	Auto Data statments for M/L
"	"	223	Shuffle 52 cards
"	"	250	M/L Numeric output (pt 3)
"	"	254	RESET RUN & Dynamic Keyboard
COMPUTE	Jul83	192	M/L Numeric Output
COMPUTE	Jul83	192	M/L Numeric output (pt 2)
"	"	221	Artifacting
COMPUTE	Sep83	209	Easy Page Flipping
"	"	210	Create Data file sys (pt III)
"	"	239	Fontbyter
COMPUTE	Oct83	232	Multicolor Char editor
"	"	290	Safe RAM for Basic
COMPUTE	Nov83	264	GTIA Textwriter
COMPUTE	Dec83	216	MLX M/L entry program (Brannon)
"	"	288	Screenbyter-Create Grapics
ANTIC	F/M83	33	Display List tut (good)
"	"	28	DOS extended directory
ANTIC	Apr83	34	Trace utility
"	"	48	Text window for GTIA
"	"	80	Basic using M/L
ANTIC	May83	6	No INPUT "?"
"	"	10	Recover Double name disk files
"	"	28	Using Sector 720
ANTIC	Jun83	24	DLI Tutorial
"	"	41	Data Base Basics
ANTIC	Jul83	10	Recover double name Disk files
ANTIC	Aug83	22	3D
"	"	57	Page Flipping
ANTIC	Sep83	37	Using M/L in Basic
AT CONN	SUM83	26	Date,Dollor,Varnames,Disklook-UTL
"	"	27	Harmony (Hudson)
"	"	56	Cartoon Animation
"	"	49	Storing Data on Disk
"	"	56	Char Set Cartooning
AT CONN	FAL83	15	Using OPEN to creat Disk file
ANALOG	#10	100	Fillerup W/Source code (Hudson)
"	"	73	Moving Players BAS&M/L (Hudson)
"	"	94	Installing GTIA
ANALOG	#11	29	Moving Missles Bas&M/L (Hudson)
"	"	35	Strings in Atari Basic
"	"	98	GR 7 Handler M/L (Hudson)
ANALOG	J/A83	24	Save Micropainter (Hudson)
"	"	30	M/L GTIA GR demo
"	"	43	Letter Perfect tutorial
"	"	48	Microsoft Corrections
"	"	84	M/L String search
"	"	111	Livewire W/Source (Hudson)
ANALOG	S/O83	66	M/L BOOT CAMP PT1-HEX/DEC/BIN (Hudson)
"	"	103	CIO and Basic
"	"	113	Fine Scrolling (Pt I)
ANALOG	N/D83	25	Fine Scrolling (pt II)
"	"	80	RETTROFIRE Source (Hudson)
"	"	37	HEXPAD-add DATA statements to Basic
"	"	111	mUse-Basic mem monitor (Moriarty)
"	"	125	M/L BOOT CAMP PT2-Addressing
CR.COMP	Feb83	266	Rainbow Title Utility

CR.COMP	Sep83	284	Unlistable BASIC
			******* 1984 *******
COMPUTE	Jan84	186	Self Relocatable M/L (Wilkinson)
COMPUTE	Feb84	162	Formatted Screen I/O (Wilkinson)
ANALOG	Jan84	104	Fine Scrolling (pt III)
"	"	124	M/L BOOT CAMP PT3-BCD
"	"	131	Index to ANALOG =LOG
ANTIC	Jan84	94	Page Flipping in M/L (Plotkin)
AT CONN	SPR84	57	Using TRACKBALL
"	"	62	Fun with Strings
AT CONN	WTR84	46	Dr Wacko Sound UTL
"	"	53	String loading & Data to ATASCII
"	"	57	ANTIC IC Mode 4 Characters
AT CONN	SUM84	65	Auto RETURN using poke 842,13
COMPUTE	Mar84	139	Relational Operators
"	"	155	Binary Loader from Basic
"	"	165	Keyboard Buffer
COMPUTE	Apr84	143	Using DOS Copy File with Basic
ANALOG	Feb84	16	Stars 3-D using M/L
"	"	108	M/L BOOT CAMP PT4-Branching,Indexing
ANALOG	Mar84	96	M/L BOOT CAMP PT5-Add,Subtract
ANTIC	Feb84	89	Color fill with M/L
"	"	15	All about Atari OS PT1
ANTIC	Mar84	18	All about Atari OS PT2
ANTIC	Apr84	50	Creative PM GR.-Add Color
"	"	82	Note and Point
"	"	90	Safe RAM for M/L
COMPUTE	MAY84	10	Disable break key
COMPUTE	Jun84	98	Binary object file loader in M/L
"	"	126	Gr.0 text in 4 colors
"	"	129	Atari TABS (4 ways)
COMPUTE	Jul84	112	Decimal mode,what is BCD?
"	"	140	Macrodos Source code
COMPUTE	Sep84	116	Bulldozer Sort
COMPUTE	Nov84	177	Atari easy scroll
ANALOG	Apr84	68	Using Vectors (Hudson)
ANALOG	Jul84	92	SCREDIT Scrolling screen editor
ANALOG	Sep84	72	Sound FX Create Sound effects
ANALOG	Oct84	16	Alternate Cursor
"	"	57	GR 8 Char Generator
ANALOG	Dec84	65	How BASIC works
ANTIC	Jun84	41	Scrolling short coarse
"	"	48	More PM tutorial
"	"	87	Graphics Macros for M/L
ANTIC	Jul84	21	AMODEM Basic source (Steinbrecher)
ANTIC	Aug84	21	HORSPLAY-GREAT!(Elwood)
"	"	48	Boolean in BASIC
ANTIC	Sep84	64	Rotate your player (M/L)
ANTIC	Oct84	38	Bouncing ball-PM by Lucasfilm guy
"	"	42	ANIMATE.BAS-Antic 4/5 Editor
ANTIC	Nov84	90	PMCURSOR.BAS-Create a cursor(White)
			******* 1985 *******
COMPUTE	Jan85	138	M/L Multiplication
COMPUTE	Feb85	109	Adding sound effects
COMPUTE	Mar85	107	Disk RX
COMPUTE	Mar85	120	Auto DATA line generation

COMPUTE	Apr85	112	M/L String creator
ANALOG	Jan85	37	Painles$ P/M mover
ANALOG	Apr85	30	Extended calculations
ANALOG	May85	46	Z-Plotter
"	"	62	Boulder Bomber source
"	"	85	TWO-FIFTY-SIX -Select Color Utility
ANTIC	Mar85	12	KWIK DUMP-Screendump and print graphics tut
COMPUTE	Jul85	10	Text adventure Vocabulary
COMPUTE	Aug85	68	Atari color mixing
ANALOG	Jun85	71	"V:" Device -mini RAM storage
ANALOG	Jul85	29	Translator-change OS in RAM
ANALOG	Aug85	83	Sound TUT for M/L programmers
ANTIC	Jun85	14	Pixel averaging
"	"	38	3-D Rotate and zoom

INDUSTRIAL ATARI

Lee S. Brilliant, MD LA-ACE

In this often Ho-Hum world of computers where MS-DOS is the only language spoken and where excitement is defined as a new clock speed or a box that has a slightly different shade of beige, occasionally something happens to reaffirm my faith in the diversity of man. Such an event took place the first weekend in April. It was a beautiful spring day as I drove up to the front doors of the Hotel Intercontinental in San Diego. I was there to learn something new and exciting in the field of Obstetrics and as I entered the lobby, the doorman's white-gloved hand opened 3/4 inch thick solid glass doors. This beautiful new hotel is located on the bay in San Diego; the basic room rate starts at $130 a night! There, clearly visible from across the lobby was a 25 inch color monitor displaying the entire day's conference schedule in a beautifully scrolling text display. I was struck instantly by it. It was pale blue on dark blue, 20 columns by 12 rows in easily readable blocky letters 6 by 6 pixels in an 8 by 8 matrix. And the character set looked amazingly familiar! Could it be? Naw, not at a place as obviously sumptuous as this one!

Through the concierge I contacted the director of conference services to inquire about their system. She graciously showed me the control center from which 8 displays were run. The one in the main lobby scrolls vertically and shows the whole day's meeting schedule while outside of each conference room is a built-in monitor displaying only the activity in that room. The main control is your standard off the shelf low end generic IBM which connects to a fascinating large blue box: obviously custom hardware. I didn't have the nerve to ask her to pull the cover off and so I left, disappointed that it wasn't an 800 sitting there running the show. Later that day I passed by the same desk and noted the cabinet was unlocked and that no one was around. The temptation was too great so I opened the cabinet to gaze at that blue mystery. Aha, divine inspiration: there were no screws holding the cover in place! I held my breath, looked around again, and pulled the cover off. Lo! Inside was a nice neat stack of 8 circuit boards that were hauntingly familiar, there were the guts of 8 600XLs lock, stock and without keyboards or cases. I quickly put everything back together and then nonchalantly wandered off, deeply satisfied.

Later I found out who manufactured this system and contacted the creator Michael F. Young from Young Design in Tysons Corner, Virginia and he told me a little about his system. It is a 128 channel data interface and multiplexer which he built. It can connect up to 128 ATARI computers to the serial output of an IBM and feeds data bytes to the XLs via their joystick ports. Video outputs are then routed to the hotel's displays. Custom software is installed in the XL's by replacing their BASIC ROMs with 2764 EPROMs and modifying the circuit boards slightly. When the operating system goes to initialize BASIC, it starts the custom software instead.

So far Young Design's system (which was sold and marketed through TEC Communications to Hotel Intercontinental) is a one-of-a-kind, although Mr. Young would hope to sell more. When I asked him how it was that he used ATARI computers, his answer was "because of its graphics capability and low cost." The system is capable of much more than I saw but the conservative hotel did not want four color screens. Mr. Youn's first use of ATARI was a system for video paging at conventions using a single ATARI to run several monitors on the convention floor. He used a keyboard, connected via a UHF link, to allow messages to be changed from anywhere on the convention floor. "The system worked well but there was not much market for it." Hotel officials saw the set-up thus leading to the development of the system I saw.

Adventist Hospital in Glendale uses an 800 with a similar scrolling display to replace the typical bulletin board in alerting physicians to hospital activities. The humble ATARI is more than just a game machine, it is an industrial workhorse, the rest of the world just doesn't know that. As Mr. Young said during his interview "Don't tell them they were Atari computers, please I hope you didn't tell them!" I did. The real question, though, is why I spend my time investigating such seemingly unimportant matters? Well beside my obvious interest in Atari computers, it rained two of my three days in San Diego! What else could I do?

What's New in the 16-bit Public Domain

by Ray Maynard

----- DCOPY20 -----

DCOPY is a multipurpose file maintenance utility that should meet almost anyone's needs. It is a shareware product (meaning that you are free to distribute it and anyone who finds it useful is requested to send the author a donation) by Ralph Walden of Eugene ACE. Some of the earlier versions of DCOPY seemed to have (or create) some problems but this version has worked very well.

DCOPY is completely menu driven and is thus very easy to use. Each menu option is usually self-explanatory. However, the proper use of some of the parameters is somewhat obscure and a thorough reading of the disk-resident documentation that accompanies the program is mandatory.

I have found DCOPY to be very useful. Most disk operations are performed much faster using DCOPY than using the mouse and desktop. DCOPY was downloaded from the AtariBase BBS.

----- ELVERT Version 0.06 -----

ELVERT is a file conversion utility written (and copyrighted) by Dan Rhea, CompuServe SYSOP of the ATARI16 forum. Although copyrighted, Dan allows its free distribution and use but ELVERT must not be sold.

ELVERT provides many of the conversion capabilities of DCOPY plus a few more.

------ DISKEDIT -----

DISKEDIT is a disk sector editor/dumper. It displays sectors in both ASCII text and hex numbers.

----- LOOKSE -----

LOOKSE is a program that displays the 16-bit internal value of each key that you press. The value is shown in both hex and decimal formats. This program can also be used to check if your keyboard has the "SE" bug which existed in some early

ST keyboard (mine has the @$%^@# bug!).

LOS ANGELES ATARI COMPUTER ENTHUSIASTS

LACE

----- STARTUP Version 1.00 -----

STARTUP.PRG is a Batch Startup program by Murray Levine that executes commands from the file STARTUP.INF upon booting your ST. STARTUP.PRG must be placed in the \AUTO\ folder on the boot drive (usually A). The program first checks for file

STARTUP.INF in the same directory and, if it's not found there, tries the main directory (ex. A:STARTUP.INF). Once you start using STARTUP.PRG, these two files can be the only files needed in the AUTO directory because all other files to

be executed can be located in other drives and folders and can even have parameters passed to them. The order in which files are written to the \AUTO\ folder is no longer important since STARTUP executes commands as they are encountered in

file STARTUP.INF.

STARTUP.PRG supports a few UNIX-like commands as well as some special commands that are used in the STARTUP.INF file.

----- TOUCH -----

TOUCH is a program that updates the creation date and time of all files that meet the file specification that is given as a command argument. This is used in order to "trigger" the dumping of a file during a backup dump that would ordinarily

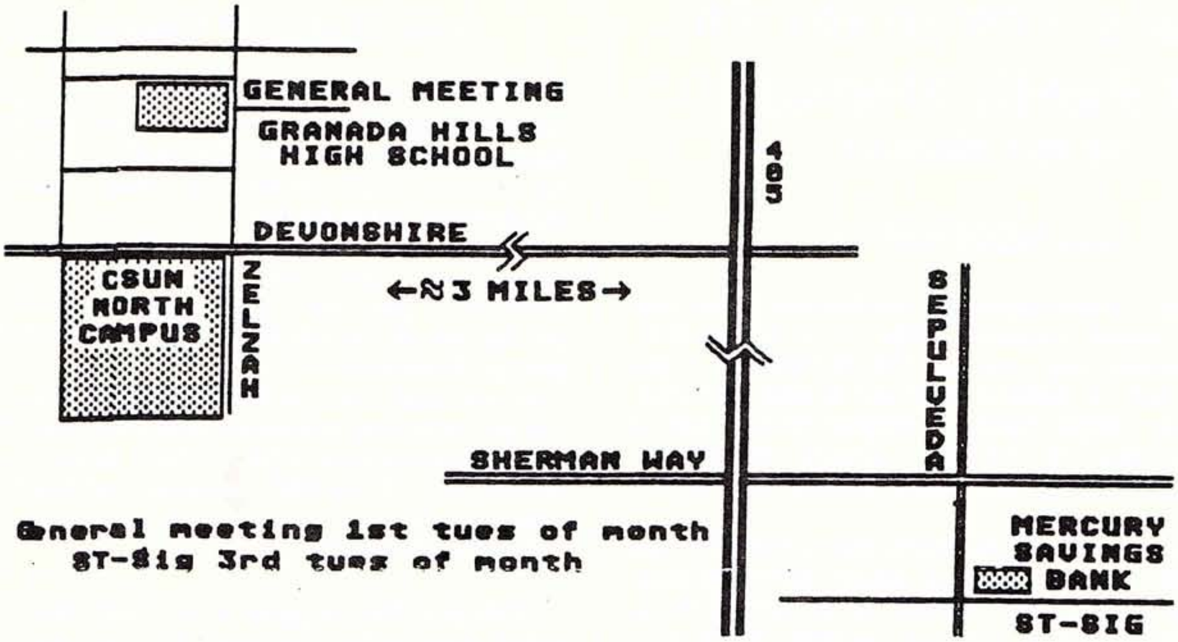
not be dumped. TOUCH is most useful for hard disk owners. The ARC file contains both the .PRG file and C source code.

There's a lot more public domain stuff out there...so much that I can't keep up with it. I'll keep trying though, and will try to keep you posted.

USERS Helping USERS

Douglas Kelley **(818) 349-5006**	**Programming Basic, Pascal, Fortran** **2pm to 10pm Everyday**
Jon DeKeles **(818) 363-5660**	**Programming Basic, Modems** **6pm to 9pm Mon - Fri**
Ed Davenport	**8-bit (818) 884-4695**
Lee Brilliant,MD **(818) 368-3522**	**8-bit hardware, interfacing,** **Advanced BASIC, Intermediate 6502**
Bob Keimach **Wordprocessing (818) 988-8366** **6pm - 6am Everyday**	**8-bit hardware, software,**

If you know a little about something or just think you do and want to help, then give your name, area of interest and phone number to Jo Anne DeKeles or Jon DeKeles to be included on the U.H.U. List.



General meeting 1st tues of month
ST-Sig 3rd tues of month

SLCC NEWSLETTER EXCHANGE
P.O. BOX 1506
SAN LEANDRO,CA 94577

LA- ACE
P.O. Box 7752
Van Nuys, CA
91409

BULK RATE
U.S. Postage Paid
VAN NUYS, CA
Permit Number 949